# What are the Differences Between the Intel® Itanium® Processor and Intel® Extended Memory 64 Technology?

Intel® Software Development Products

Programmer's Paradise®

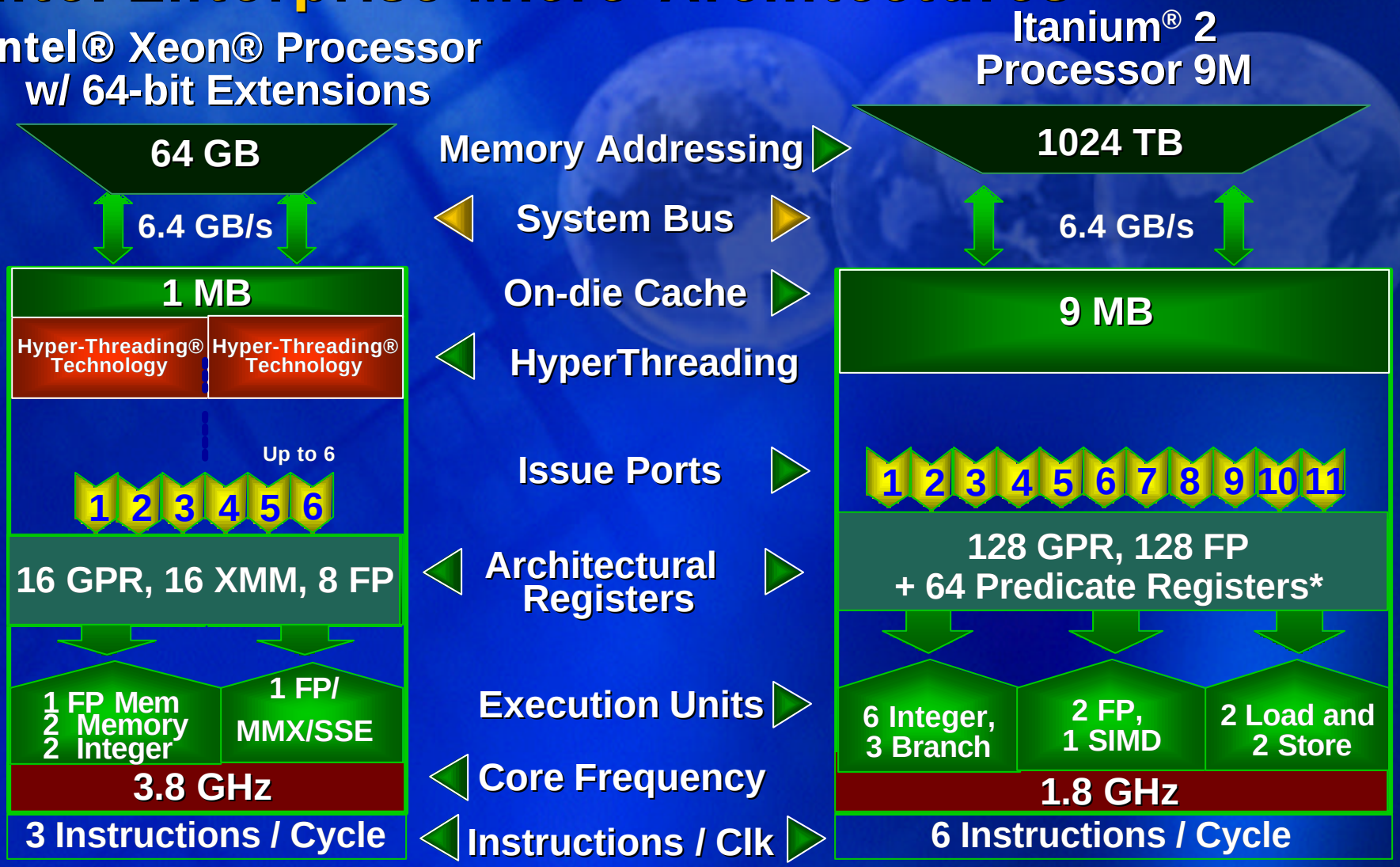
# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Key Differences:
## Itanium® processor-based systems vs. Intel® Extended Memory 64 Technology (Intel EM64T)

- Architecture
  - Parallelism:
    - Intel EM64T exploits parallelism implicitly by traditional OOO execution
    - Itanium Processor Family (IPF) exploits parallelism by in-order execution of an explicit parallel instruction stream (EPIC)
  - Compiler calls all the scheduling shots for Itanium® Processor
  - Different Instruction Set Architecture (ISA)
    - Intel EM64T uses IA32 with 64-bit extensions
    - IPF uses IA-64 ISA
  - Different architectural features supported
  - Different number of registers supported

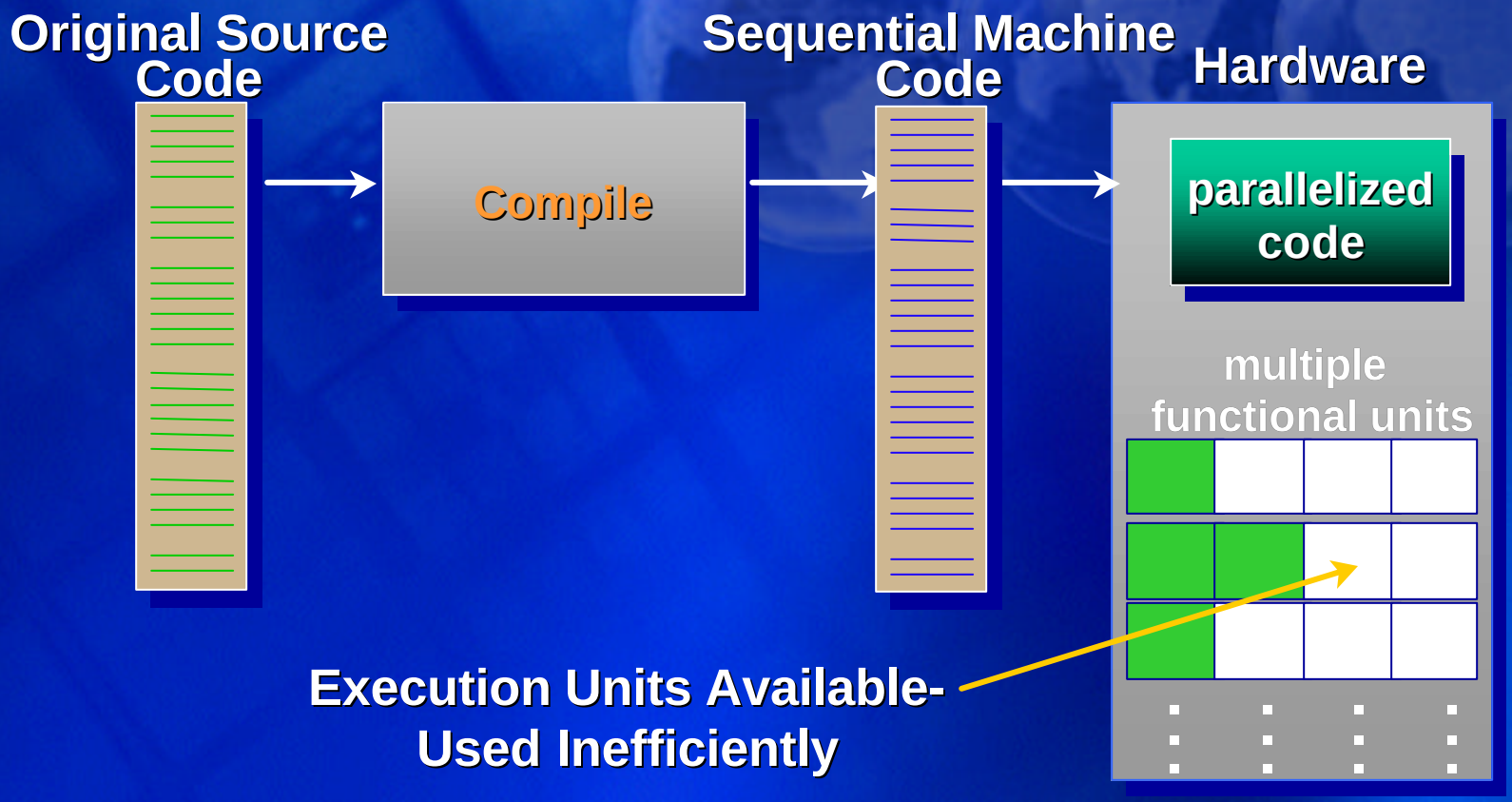
# Intel Enterprise Micro-Architectures




* Intel's EPIC technology includes 64 single-bit predicate registers to accelerate loop unrolling and branch-intensive code execution

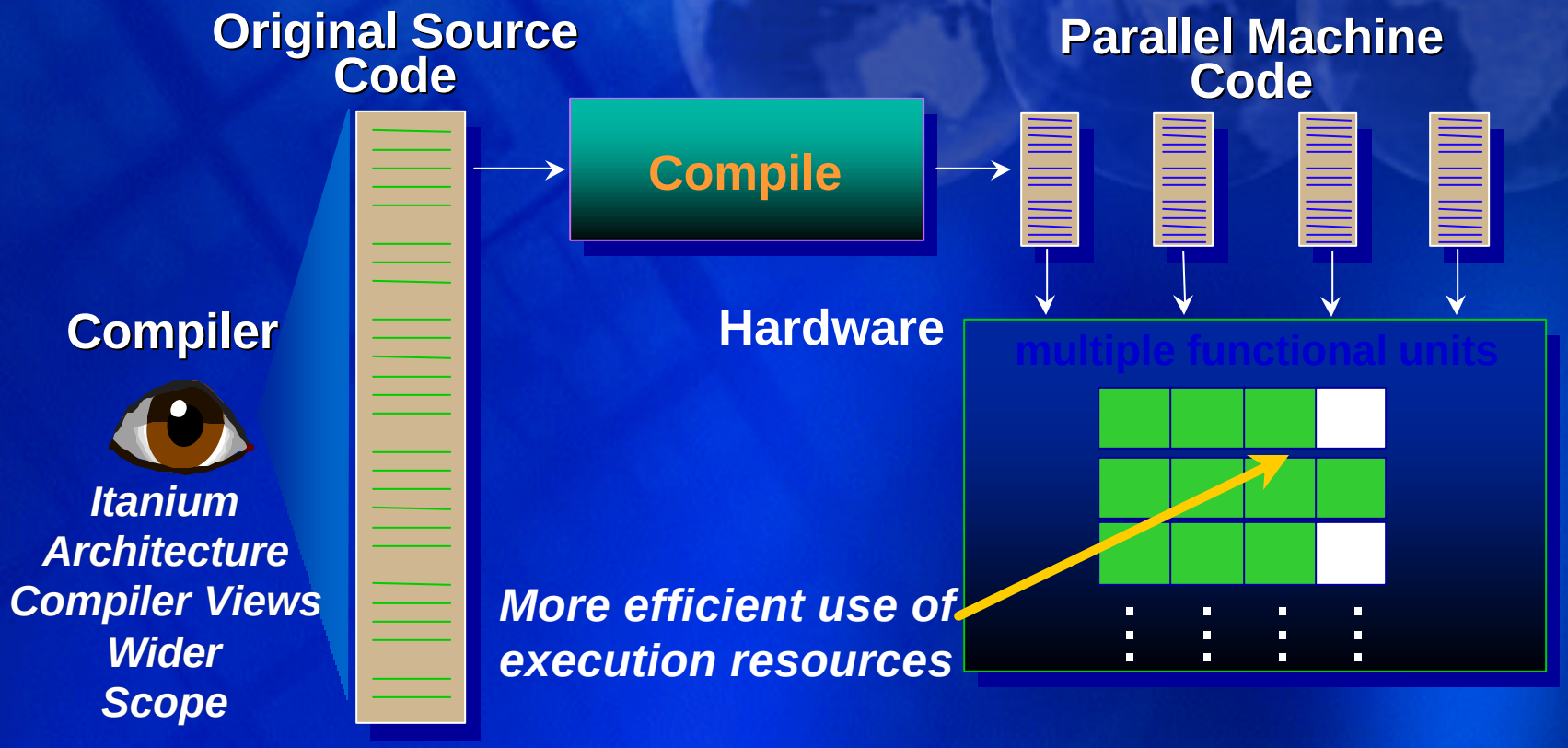
# Intel® EM64T Architectures:

## OOO Execution of a Sequential Code Stream



# Itanium® Architecture:

## Explicit Parallelism in Instruction Stream




## Increases Parallel Execution

# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Intel® C++ Compilers 8.1

- **Features:**
  - Better performance, usability, stability, quality
  - GCC 3.2, 3.3 & 3.4 C/C++ object compatibility & interoperability (C++)
  - Eclipse* integration on Linux32 (C++)
  - SLES9 & kernel 2.6 support on Linux*
- **Intel EM64T support:**
  - x86-64 OS ABI compatible/interoperable with Microsoft* and GNU* compilers/tools

- ☑ Windows
- ☑ Linux
- ☑ IA-32
- ☑ Itanium 2
- ☑ Intel EM64T
- ☑ PCA (Intel C++ for Windows)

# Supported Linux* Platforms/ Binary Compatibility

download.intel.com/software/products/compilers/techtopics/Linux_Compiler_and_GNU.pdf

| Compiler for Linux* version | Processor Architecture | Supported glibc versions | Supported kernel versions | Supported g++ versions for binary compatibility |
|---|---|---|---|---|
| 8.1 | EM64T | 2.2.4, 2.2.5, 2.3.2 | 2.4.x, 2.6.x | 3.2.x, 3.3.x, 3.4.x |
| 8.1 | Itanium® Architecture | 2.2.4, 2.2.5, 2.3.2 | 2.4.x | 3.2.x, 3.3.x, 3.4.x |
| 8.1 | IA-32 | 2.2.4, 2.2.5, 2.3.2 | 2.4.x, 2.6.x | 3.2.x, 3.3.x, 3.4.x |

# Object Compatibility on Windows*

- Intel compiler-generated binaries will be binary compatible with Microsoft compiler-generated binaries for Intel EM64T
  - When Microsoft Windows* OS debuts for Intel EM64T platforms
- Already binary-compatible with Microsoft Windows binaries on Itanium Processor family and IA-32 platforms

# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality:
  - Intel® EMT64
  - Itanium® processor-based systems
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Compiler functionality:

## Intel® Extended Memory 64 Technology

- Comparison to IA-32 “classic” Intel® compiler
  - Compiler is very similar to the IA-32 compiler
  - Supports SSE, SSE2, SSE3
  - Has double registers - 16 XMM and 16 GP registers
  - Extension to IA-32 Instruction Set Architecture
- Separate binary from Intel® 8.1 “classic” compiler
  - Separate binary from Itanium Compiler also
  - Free download after you have acquired the 8.1 “classic” compiler
- Memory Model support
  - Small (default)
    - Code and data is restricted to the first 2GB of address space.
  - Medium
    - Code is restricted to the first 2GB, no restriction on data.
  - Large
    - No restrictions on code or data.

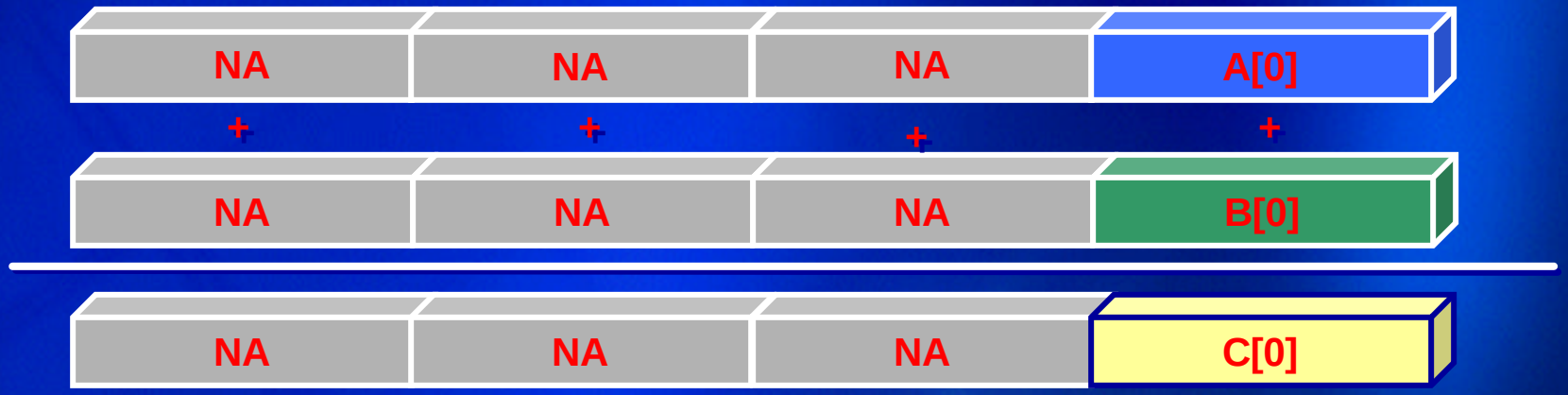
# Vectorization Converts Loops

## Intel Extended Memory 64 Technology

- Example Prior to Vectorization:

```
for (I=0;I<=MAX;I++)
                c[I]=a[I]+b[I];
```

- Traditional Architecture:



# Vectorization Converts Loops

## Intel Extended Memory 64 Technology

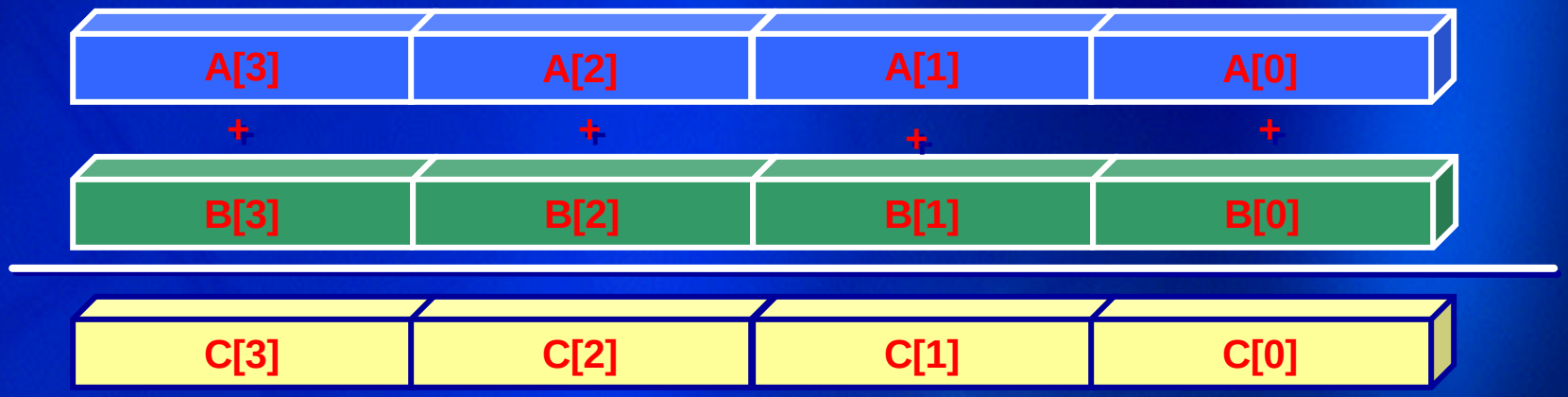
- Example with Vectorization:

```
for (I=0;I<=MAX;I++)
    c[I]=a[I]+b[I];
```

- Usage:

(Linux*) -[a]xW, -[a]xP

(Windows*) -Q[a]xW, -Q[a]xP



# Compiler Vectorization Reports:

## Intel Extended Memory 64 Technology

- **Availability:** Report information enabled with any x, ax, Qx, Qax vectorization switch
- **Information:**
  - vec_report0: no diagnostic information
  - vec_report1: loops successfully vectorized
  - vec_report2: Adds loops not vectorized
  - vec_report3: Adds info dependencies – proven or assumed
- **Syntax:**
  - **-vec_report*x***
- **References:**
  - www.intel.com/software/products/compilers/techtopics/perf_guide.pdf

# Recommended Intel EM64T Options
## (These only apply to the Intel compilers for EM64T)

| Target | Linux* | Windows* |
|---|---|---|
| Only Intel EM64T & generate SSE3 where possible | xP | QxP |
| Intel EM64T & SSE3, yet will run on any Intel EM64T-compatible system without SSE3 (uses CPU dispatch) | axP | QaxP |
| Intel EM64T and any Intel EM64T-compatible system (not using CPU-dispatch). Will not utilize SSE3, and may not be as optimal as “axP” or “xP” | xW | QxW |

# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality:
  - Intel® EMT64
  - Itanium® processor-based systems
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Compiler functionality:
## Itanium-based systems

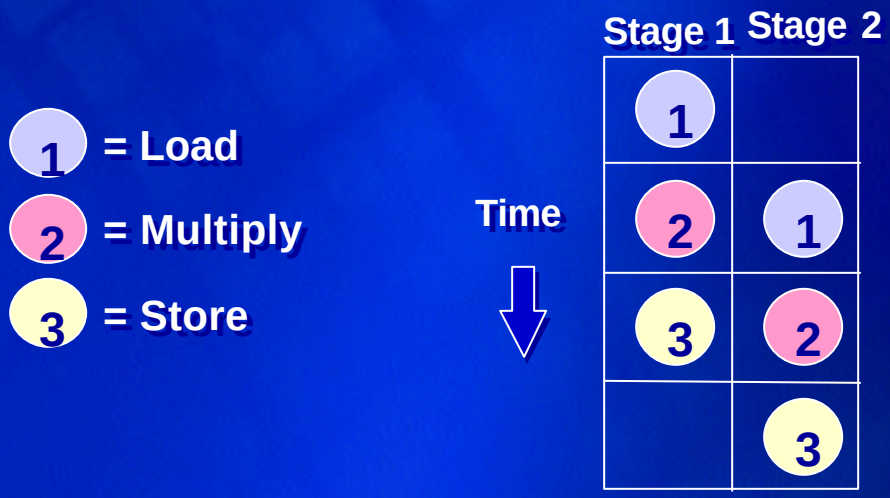
- Instruction Set Architecture
  - Based on Itanium® Instructions
- Separate binary from 8.1 “Intel EM64T” Compiler
  - Separate binary from IA-32 Compiler also
- Compiler generates ‘Explicit Parallel Instruction Stream’ ( grouping instruction to sets which can be issued at the same time)
- Many key architecture features that differ from IA-32 and Intel EM64T

# Key Architectural Features:
## Itanium-based systems

- Instruction-level parallelism
  - Multiple (6 for Itanium® 2) instructions retired per clock
- Large number of registers (over +324 )
- Speculation
- Predication
- Register stack
- Software pipelining

# Software Pipelining:
## Itanium-based systems

- Modulo-scheduled loops
  - *Software pipelining* allows several instructions in various loop stages to be executed at a given time.
  - Use special branch loop instructions, application registers, predication and rotating registers.




**Improves Loop Performance While Exploiting Parallelism**

# Compiler Optimization Reports:

## Itanium-based systems

- **Availability:** Report information enabled at -O2 and above
- **Information:**
  - ecg_swp: Did a given loop Software Pipeline and if not why not
  - hlo: What prefetching & loop transformations did compiler apply
  - ipo: What functions were inlined
- **Syntax:**

  -opt_report (writes to stderr)

  -opt_report_file“filename”

  -opt_report_phase <ecg_swp|hlo| ecg| ipo|…>

  -opt_report_routine <name>

  -opt_report_help
- **References:**
  www.intel.com/software/products/compilers/techtopics/perf_guide.pdf

# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Targeting Address Spaces Larger than 2GB

- Itanium-based systems
  - Compile application with Intel Compiler for Itanium
    - Native compilation on Linux*
    - Native & Cross compilation on Windows*
- Intel EM64T systems
  - Compile application with Intel® EM64T Compiler
    - Native compilation on Linux
    - Cross compilation on 32-bit Windows or run as 32-bit on Intel EM64T
  - Use Medium or Large memory model switches

# Targeting addresses larger than 2GB with EM64T Compiler

- Considerations
  - Need to build with one of –mcmodel=medium or –mcmodel=large
  - Also need –i_dynamic to address 2GB or more of data
  - It's essential to specify both the mcmodel and i_dynamic switches to get long addresses

# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Porting to 64 Bits

- Root of the problem
  - Moving from 32-bit to 64-bit processing requires 64-bit pointers
  - Some legacy apps treated integers, longs & pointers as equivalent
  - Pointers may accidentally be truncated
- Use compiler warning switch
  - **Wp64**
  - Warns of potential 64-bit pointer issues
- Porting to 64 bits is not a new concept
  - See references

# Data Models

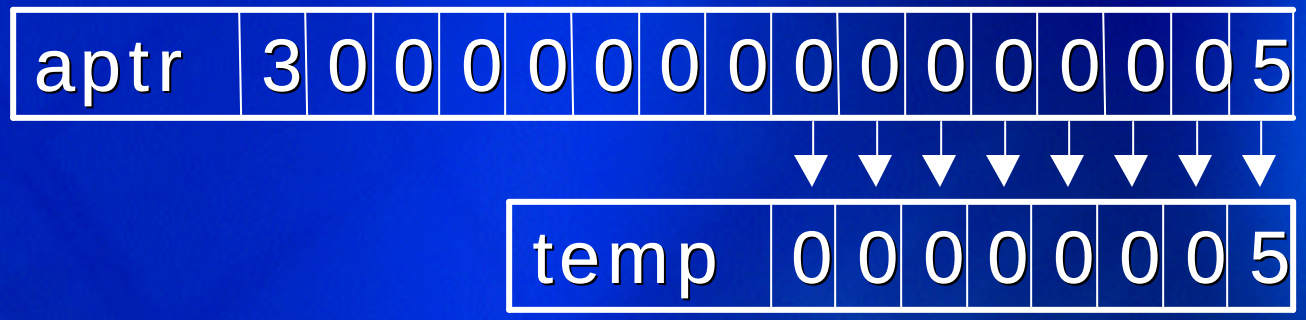
- ILP32
  - int, long and ptr are 32 bits.
  - They are used by 32-bit operating systems.
- LP64
  - int is 32 bits.
  - Long and pointer are 64 bits.
  - This is used by 64-bit UNIX* & 64-bit Linux*.
- P64 (or LLP64)
  - int and long are 32 bits; pointer is 64 bits.
  - This is used by Windows*.

| | ILP32 size (bits) | LP64 size (bits) | P64 size (bits) |
|---|---|---|---|
| int | 32 | 32 | 32 |
| long | 32 | 64 | 32 |
| pointer | 32 | 64 | 64 |

# Pointer Truncation

- Root cause of most porting issues

```
void *aptr;
int temp;
aptr=malloc(sizeof(void *));
temp=aptr;
//truncation!
```



# Agenda

- Key Architectural Differences
- Intel® Compiler Features
- Intel Compiler Functionality
- Targeting Address Spaces Larger than 2GB
- Porting to 64 Bits
- More Information

# Intel® Software College

## Get the training you need

## Bring out your application performance…

- **64-bit computing: Optimizing Applications on the Intel® Xeon™ Processor**
  - Intensive 3-day, instructor-led lecture and hands-on labs
  - Improve performance through Intel® VTune™ Performance Analyzer, Intel® compilers, libraries, and multithreading tools.
- **Advanced Optimization for Intel® Extended Memory 64 Technology (Intel® EM64T)**
  - FREE online audio enabled presentation
  - Review optimization guidelines, large memory considerations, and software tools

**Register at: www.Intel.com/software/college**

# Summary

- Intel® 8.1 Compilers support both Intel® EM64T and Itanium® Processors
- You can now address more than 2GB of Data
- Code Ported to 64 bits can be compiled for either Intel® EM64T or Itanium® Processors

# Call To Action

- Download evaluation copies of Intel compilers today at:
  - www.intel.com/software/products
- Enroll in Intel® Software College Intel EM64T course
- Go to the Intel EM64T website
  - www.intel.com/technology/64bitextensions/

# References

- Porting
  - **“Porting Linux* Applications to 64-Bit Intel® Architecture”**
    - www.intel.com/cd/ids/developer/asmo-na/eng/167739.htm?page=1
  - **“Porting to 64-Bit Intel Architecture”**
    - www.intel.com/cd/ids/developer/asmo-na/eng/technologies/64bit/170114.htm?page=2
  - **“Preparing Code for the IA-64 Architecture (Code Clean)”**
    - cache-www.intel.com/cd/00/00/01/79/17969_codeclean_r02.pdf
  - **"Code Clean Enables Software in Both IA-32 and IA-64 Worlds"**
    - developer.intel.com/update/issue/issue23.pdf
  - **"Getting Ready for 64-bit Windows*"**
    - msdn.microsoft.com/library/default.asp?url=/library/en-us/win64/win64/getting_ready_for_64_bit_windows.asp
- Binary Compatibility
  - download.intel.com/software/products/compilers/techtopics/Linux_Compiler_and_GNU.pdf

# References

- Intel® EM64T websites
  - **Intel® Extended Memory 64 Technology FAQ's**
    - www.intel.com/technology/64bitextensions/faq.htm
  - **Intel® Developer Services Software and Solutions, Intel EM64T**
    - www.intel.com/cd/ids/developer/asmo-na/eng/index.htm
- Intel® Early Access Program
  - www.intel.com/cd/ids/developer/asmo-na/eng/eap/19383.htm
- Developer's Guides
  - **Intel Extended Memory 64 Technology Software Developer's Guide**
    - ftp://download.intel.com/technology/64bitextensions/300835.htm
  - **Itanium® Architecture Software Developer's Manual**
    - www.intel.com/design/itanium/manuals/iiasdmanual.htm
  - **IA-32 Intel® Architecture and Intel Extended Memory 64 Software Developer's Manual Documentation Changes**
    - www.intel.com/design/pentium4/specupdt/252046.htm
- Optimization & Vectorization reports
  - www.intel.com/software/products/compilers/techtopics/perf_guide.pdf
  - www.presentationselect.com/hp/archive.asp?EventID=428&AT=WMStreamArchiveURL

Programmer's Paradise®

# Backups